UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/689,126	10/20/2003	John B. Condon	BLD920030025US1	2905

50441 7590 07/24/2007
DUFT BORNSEN & FISHMAN, LLP
1526 SPRUCE STREET
SUITE 302
BOULDER, CO 80302

EXAMINER	
DICKERSON, CHAD S	

ART UNIT	PAPER NUMBER
2625	

MAIL DATE	DELIVERY MODE
07/24/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/689,126	CONDON ET AL.	
	Examiner	Art Unit	
	Chad Dickerson	2625	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *20 October 2003*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-24* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-24* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *10/* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *see attachment*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Claim Rejections - 35 USC § 103

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

2. Claims 1-24 are rejected under 35 U.S.C. 103(a) as being unpatentable over McIntyre '478 (US Pat No 6690478) in view of Wood '934 (US Pub No 2004/0243934).

Re claim 1: McIntyre '478 discloses a method and apparatus for utilizing multiple versions of a page descriptor language comprising the steps of:

b) processing each of the plurality of work units by at least one compute node to convert each work unit into a second format (i.e. in McIntyre '478, a plurality of print jobs, considered as work units can be processed. The printer driver (114), considered as the compute node, processes the incoming print jobs by recognizing the type of PDL is input into the system. The printer driver selects a PDL type from a PDL registry (112) to correspond with the incoming data and this PDL type chosen is used to convert the data into a different format, analogous to the second format, which is a low-level data stream; see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25).

However, McIntyre '478 ails to teach a) parsing the datastream into a plurality of work units in a first format,

However, this is well known in the art as evidenced by Wood '934. Wood '934 discloses a) parsing the datastream into a plurality of work units in a first format (i.e. the apparatus and method of Wood '934 divides the PDL data stream to provide a plurality of PDL segments and then to create associated first and second data segments for each of the PDL segments. These segments are in a PDL format, analogous to a first format and the function of parsing is performed by the dividing and interpreting the PDL data stream into a plurality of segments, which the plurality of segments are considered as the plurality of work units; see figs. 1-4; paragraphs [0004]-[0014] and [0025]-[0029]).

Therefore, in view of Wood '934, it would have been obvious to one of ordinary skill at the time the invention was made to parse the datastream into a plurality of work units in a first format in order to divide the PDL data stream to provide a plurality of PDL segments (as stated in Wood '934 paragraph [0014]).

Re claim 2: The teachings of McIntyre '478 in view of Wood '934 are disclosed above. McIntyre '478 discloses the method, wherein the parsing step (a) includes:

(a1) providing a plurality of sources, wherein each source is associated with at least one transform (i.e. in McIntyre '478, a method for registration and selection of multiple page description languages (i.e. personalities) is presented. The personalities, analogous to a plurality of sources, are associated with a transform, or conversion, in order to convert the input instructions into a printer dependent data stream interpreted by the printing subsystem (108) to produce an output page; see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25);

(a2) instantiating at least one source of the plurality of sources, wherein the at least one instantiated source is associated with the datastream format (i.e. the printer driver (114) recognizes a realization of the personality related to the transform, or instantiates one transform, in order to perform a conversion, that is associated with received instructions. The printer driver (114) analyzes these received instructions and chooses one of the multiple PDLs registered in the PDL registry (112); see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25); and

(a3) utilizing the at least one source (i.e. once the printer driver (114) finds an appropriate version of a PDL registered within the PDL registry (112), the printer driver invokes the personality, or utilizes the personality analogous to the source, to convert the instructions from a high level language to a printer dependent data stream; see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25).

However, McIntyre '478 fails to teach to parse the datastream.

However, this is well known in the art as evidenced by Wood '934. Wood '934 discloses to parse the datastream (i.e. the apparatus and method of Wood '934 divides the PDL data stream to provide a plurality of PDL segments and then to create associated first and second data segments for each of the PDL segments. These segments are in a PDL format, analogous to a first format and the function of parsing is performed by the dividing and interpreting the PDL data stream into a plurality of segments, which the plurality of segments are considered as the plurality of work units; see figs. 1-4; paragraphs [0004]-[0014] and [0025]-[0029]).

Therefore, in view of Wood '934, it would have been obvious to one of ordinary skill at the time the invention was made to parse the datastream in order to divide the PDL data stream to provide a plurality of PDL segments (as stated in Wood '934 paragraph [0014]).

Re claim 3: The teachings of McIntyre '478 in view of Wood '934 are disclosed above. McIntyre '478 discloses the method, wherein the processing step (b) includes:

(b1) loading the at least one transform associated with the at least one instantiated source in the at least one compute node (i.e. the printer driver (114) recognizes a transform, or conversion, associated with at least one realization of the personality, analogous to an instantiated source, that is able to perform the transform and invokes the personality, or PDL, to perform the conversion of the received instructions. The action of invoking the personality after associated the PDL with the received data is analogous to immediately loading the transform in order to be utilized for transformation of the received data; see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25); and

(b2) utilizing the at least one transform to convert a work unit of the plurality of work units from the first format to the second format (i.e. in the system of McIntyre '478, the system is able to process a plurality of print jobs, considered as work units. The PDL utilized by the printer driver (114) to convert the received instructions, or print job, from a high level language to a printer dependent data, or language, is analogous to

converting a work unit from a first format to a second format; see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25).

Re claim 4: The teachings of McIntyre '478 in view of Wood '934 are disclosed above.

However, McIntyre '478 fails to teach the method further comprising: (c) load balancing the plurality of work units.

However, this is well known in the art as evidenced by Wood '934. Wood '934 discloses the method further comprising: (c) load balancing the plurality of work units (i.e. by sequentially assigning data files in Wood '934 to the next available PDL processor, the workload of processing data files (20) and (22) may be automatically load-balanced between various PDL processors. The data files are analogous to the plurality of work units; see paragraph [0033]).

Therefore, in view of Wood '934, it would have been obvious to one of ordinary skill at the time the invention was made to have the step of load balancing the plurality of work units in order to automatically load-balance the workload of processing data files in the system (as stated in Wood '934 paragraph [0033]).

Re claim 5: The teachings of McIntyre '478 in view of Wood '934 are disclosed above.

However, McIntyre '478 fails to teach the method wherein the load balancing step includes: (c1) generating at least one queue for the plurality of work units; and (c2) distributing each work unit from the at least one queue to the at least one compute nodes in an order.

However, this is well known in the art as evidenced by Wood '934. Wood '934 discloses the method wherein the load balancing step includes:

(c1) generating at least one queue for the plurality of work units (i.e. the memory buffers (40), are associated with the PDL processors (38) and the buffers stores the data files (20 and 22). Several data files are queued up for each processor, so that no PDL processor runs out of data files to process; see figs. 3-5; paragraphs [0030]-[0034]); and

(c2) distributing each work unit from the at least one queue to the at least one compute nodes in an order (i.e. the data files (20 and 22), considered as the work units, are distributed to the PDL processors (38) from the buffer memories, which act as queues in the system. The PDL processors (38) in Wood '934 are analogous to computer nodes. The data files are distributed by the scheduling process (24), which accounts for the order-dependence of the data files; see figs. 1 and 3-5; paragraphs [0030]-[0034]).

Therefore, in view of Wood '934, it would have been obvious to one of ordinary skill at the time the invention was made to generate a queue for a plurality of work units and distribute each work unit from the queue to a computer node in an order in order to assign data files from buffer memories, which have several data files queued up, to PDL processors while taking the order-dependence of the data files into account (as stated in Wood '934 paragraphs [0031]-[0034]).

Re claim 6: The teachings of McIntyre '478 in view of Wood '934 are disclosed above.

McIntyre '478 discloses the method further comprising:

(d) returning the plurality of processed work units from the at least one compute node to the at least one source (i.e. in McIntyre '478, the printer driver (114) invokes the personalities to perform the transformation of receiving information. If the system invokes a personality to perform the conversion of input received data, then the personality aspect of the invention performs the transformation of the PDL and returns this conversion to the printer driver (114). In this case, the personality utilized to perform transformation is considered to be the compute node, which performs the transformation, and the printer driver (114) is considered to be the source since the data was received by the printer driver (114) and sent to the personality (106) from the printer driver (114) and sent back to the printer driver (114) in order to be output by the output device; see figs. 1 and 5; col. 2, lines 18-30, col. 3, lines 1-67, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25).

However, this is well known in the art as evidenced by Wood '934. Wood '934 discloses the method further comprising:

in the order it was distributed (i.e. Wood '934 reveals that the prior art involves a process of combining a data stream portion in the order of the original PDL data stream. In Wood '934, when the PDL processes the data, the scheduling process (24) accounts for the order-dependence of the data files in the buffer memories; see fig. 4; paragraphs [0030]-[0037]).

Therefore, in view of Wood '934, it would have been obvious to one of ordinary skill at the time the invention was made to return the plurality of processed work units

from the at least one compute node to the at least one source in the order it was distributed in order to take the order-dependence of the data files into account (as stated in Wood '934 paragraphs [0030]-[0034]).

Re claim 7: The teachings of McIntyre '478 in view of Wood '934 are disclosed above.

However, McIntyre '478 fails to teach the method, wherein at least one of the plurality of work units is a control work unit that includes control commands for the at least one compute node.

However, this is well known in the art as evidenced by Wood '934. Wood '934 discloses the method, wherein at least one of the plurality of work units is a control work unit that includes control commands for the at least one compute node (i.e. the segment file (22) includes all PDL commands necessary to interpret the associated segment, which includes everything in the global data file associated with the segment. The PDL commands are for commanding the actions of the PDL processors (38), considered as the compute node; see fig. 1; paragraphs [0025]-[0030]).

Therefore, in view of Wood '934, it would have been obvious to one of ordinary skill at the time the invention was made to have at least one of the plurality of work units as a control work unit that includes control commands for the at least one compute node in order to have a data file include PDL commands necessary to interpret a segment related to an input datastream (as stated in Wood '934 paragraphs [0026]-[0030]).

Re claim 8: The teachings of McIntyre '478 in view of Wood '934 are disclosed above.

However, McIntyre '478 fails to teach the method, wherein the processing step (b) includes (b1) executing the control commands in the control work unit.

However, this is well known in the art as evidenced by Wood '934. Wood '934 discloses the method, wherein the processing step (b) includes (b1) executing the control commands in the control work unit (i.e. the PDL processors (38) is commanded by the data file (22) with the PDL commands in order to interpret the associated segment related to the data file (22). Therefore, the PDL executes the control commands of the data file (22) in the segment, considered as the work unit; see fig. 1; paragraphs [0025]-[0030]).

Therefore, in view of Wood '934, it would have been obvious to one of ordinary skill at the time the invention was made to have the processing step include executing the control commands in the control work unit in order to interpret the associated segment with the data file that has the control commands (as stated in Wood '934 paragraphs [0026]-[0030]).

Re claim 9: The teachings of McIntyre '478 in view of Wood '934 are disclosed above. McIntyre '478 discloses the method, wherein the at least one source is instantiated as a dynamic library (i.e. when using a transform to convert incoming data into another form, the printer driver (114) requests for a transform to convert from the incoming PDL into a low-level language for the printer to understand. This is performed by linking the incoming data to the specific personality that will perform the transformation of the data stream to the low-level format. This process is similar to a library with a collection of

subprograms used to develop other pieces of information used by the system and provide the function of transformation that is linked to a certain input language in the invention of McIntyre '478; see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25).

Re claim 10: McIntyre '478 discloses a method and apparatus for utilizing multiple versions of a page descriptor language, the program instructions for:

b) processing each of the plurality of work units by at least one compute node to convert each work unit into a second format (i.e. in McIntyre '478, a plurality of print jobs, considered as work units can be processed. The printer driver (114), considered as the compute node, processes the incoming print jobs by recognizing the type of PDL is input into the system. The printer driver selects a PDL type from a PDL registry (112) to correspond with the incoming data and this PDL type chosen is used to convert the data into a different format, analogous to the second format, which is a low-level data stream. Also, with the above actions capable of being performed on a storage medium having stored the executable instructions to implement the teachings of the invention of McIntyre '478, the above feature of a computer readable medium containing program instructions is performed; see figs. 1, 3 and 6; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160, col. 6, lines 1-25 and col. 7, lines 7-23).

However, McIntyre '478 fails to teach a) parsing the datastream into a plurality of work units in a first format.

However, this is well known in the art as evidenced by Wood '934. Wood '934 discloses a) parsing the datastream into a plurality of work units in a first format (i.e. the apparatus and method of Wood '934 divides the PDL data stream to provide a plurality of PDL segments and then to create associated first and second data segments for each of the PDL segments. These segments are in a PDL format, analogous to a first format and the function of parsing is performed by the dividing and interpreting the PDL data stream into a plurality of segments, which the plurality of segments are considered as the plurality of work units; see figs. 1-4; paragraphs [0004]-[0014] and [0025]-[0029]).

Therefore, in view of Wood '934, it would have been obvious to one of ordinary skill at the time the invention was made to parse the datastream into a plurality of work units in a first format in order to divide the PDL data stream to provide a plurality of PDL segments (as stated in Wood '934 paragraph [0014]).

Re claim 11: The teachings of McIntyre '478 in view of Wood '934 are disclosed above. The computer readable medium of claim 10, wherein the parsing instruction (a) includes:

(a1) providing a plurality of sources, wherein each source is associated with at least one transform (i.e. in McIntyre '478, a method for registration and selection of multiple page description languages (i.e. personalities) is presented. The personalities, analogous to a plurality of sources, are associated with a transform, or conversion, in order to convert the input instructions into a printer dependent data stream interpreted

by the printing subsystem (108) to produce an output page; see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25);

(a2) instantiating at least one source of the plurality of sources, wherein the at least one instantiated source is associated with the datastream format (i.e. the printer driver (114) recognizes a realization of the personality related to the transform, or instantiates one transform, in order to perform a conversion, that is associated with received instructions. The printer driver (114) analyzes these received instructions and chooses one of the multiple PDLs registered in the PDL registry (112); see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25); and

(a3) utilizing the at least one source (i.e. once the printer driver (114) finds an appropriate version of a PDL registered within the PDL registry (112), the printer driver invokes the personality, or utilizes the personality analogous to the source, to convert the instructions from a high level language to a printer dependent data stream; see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25).

However, McIntyre '478 fails to teach to parse the datastream.

However, this is well known in the art as evidenced by Wood '934. Wood '934 discloses to parse the datastream (i.e. the apparatus and method of Wood '934 divides the PDL data stream to provide a plurality of PDL segments and then to create associated first and second data segments for each of the PDL segments. These segments are in a PDL format, analogous to a first format and the function of parsing is performed by the dividing and interpreting the PDL data stream into a plurality of

segments, which the plurality of segments are considered as the plurality of work units; see figs. 1-4; paragraphs [0004]-[0014] and [0025]-[0029]).

Therefore, in view of Wood '934, it would have been obvious to one of ordinary skill at the time the invention was made to parse the datastream in order to divide the PDL data stream to provide a plurality of PDL segments (as stated in Wood '934 paragraph [0014]).

Re claim 12: The teachings of McIntyre '478 in view of Wood '934 are disclosed above. The computer readable medium of claim 11, wherein the processing instruction (b) includes:

(b1) loading the at least one transform associated with the at least one instantiated source in the at least one compute node (i.e. the printer driver (114) recognizes a transform, or conversion, associated with at least one realization of the personality, analogous to an instantiated source, that is able to perform the transform and invokes the personality, or PDL, to perform the conversion of the received instructions. The action of invoking the personality after associated the PDL with the received data is analogous to immediately loading the transform in order to be utilized for transformation of the received data; see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25); and

(b2) utilizing the at least one transform to convert a work unit of the plurality of work units from the first format to the second format (i.e. in the system of McIntyre '478, the system is able to process a plurality of print jobs, considered as work units. The

PDL utilized by the printer driver (114) to convert the received instructions, or print job, from a high level language to a printer dependent data, or language, is analogous to converting a work unit from a first format to a second format; see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25).

Re claim 13: The teachings of McIntyre '478 in view of Wood '934 are disclosed above.

However, McIntyre '478 fails to teach the computer readable medium further comprising: (c) load balancing the plurality of work units.

However, this is well known in the art as evidenced by *2. *2 discloses the computer readable medium further comprising: (c) load balancing the plurality of work units (i.e. by sequentially assigning data files in Wood '934 to the next available PDL processor, the workload of processing data files (20) and (22) may be automatically load-balanced between various PDL processors. The data files are analogous to the plurality of work units; see paragraph [0033]).

Therefore, in view of Wood '934, it would have been obvious to one of ordinary skill at the time the invention was made to have the step of load balancing the plurality of work units in order to automatically load-balance the workload of processing data files in the system (as stated in Wood '934 paragraph [0033]).

Re claim 14: The teachings of McIntyre '478 in view of Wood '934 are disclosed above.

However, McIntyre '478 fails to teach the computer readable medium, wherein the load balancing instruction includes: (c1) generating at least one queue for the

plurality of work units; and (c2) distributing each work unit from the at least one queue to the at least one compute nodes in an order.

However, this is well known in the art as evidenced by Wood '934. Wood '934 discloses the method wherein the load balancing step includes:

(c1) generating at least one queue for the plurality of work units (i.e. the memory buffers (40), are associated with the PDL processors (38) and the buffers stores the data files (20 and 22). Several data files are queued up for each processor, so that no PDL processor runs out of data files to process; see figs. 3-5; paragraphs [0030]-[0034]); and

(c2) distributing each work unit from the at least one queue to the at least one compute nodes in an order (i.e. the data files (20 and 22), considered as the work units, are distributed to the PDL processors (38) from the buffer memories, which act as queues in the system. The PDL processors (38) in Wood '934 are analogous to computer nodes. The data files are distributed by the scheduling process (24), which accounts for the order-dependence of the data files; see figs. 1 and 3-5; paragraphs [0030]-[0034]).

Therefore, in view of Wood '934, it would have been obvious to one of ordinary skill at the time the invention was made to generate a queue for a plurality of work units and distribute each work unit from the queue to a computer node in an order in order to assign data files from buffer memories, which have several data files queued up, to PDL processors while taking the order-dependence of the data files into account (as stated in Wood '934 paragraphs [0031]-[0034]).

Re claim 15: The teachings of McIntyre '478 in view of Wood '934 are disclosed above. McIntyre '478 discloses the computer readable medium further comprising:

(d) returning the plurality of processed work units from the at least one compute node to the at least one source (i.e. in McIntyre '478, the printer driver (114) invokes the personalities to perform the transformation of receiving information. If the system invokes a personality to perform the conversion of input received data, then the personality aspect of the invention performs the transformation of the PDL and returns this conversion to the printer driver (114). In this case, the personality utilized to perform transformation is considered to be the compute node, which performs the transformation, and the printer driver (114) is considered to be the source since the data was received by the printer driver (114) and sent to the personality (106) from the printer driver (114) and sent back to the printer driver (114) in order to be output by the output device; see figs. 1 and 5; col. 2, lines 18-30, col. 3, lines 1-67, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25).

However, this is well known in the art as evidenced by Wood '934. Wood '934 discloses the method further comprising:

in the order it was distributed (i.e. Wood '934 reveals that the prior art involves a process of combining a data stream portion in the order of the original PDL data stream. In Wood '934, when the PDL processes the data, the scheduling process (24) accounts for the order-dependence of the data files in the buffer memories; see fig. 4; paragraphs [0030]-[0037]).

Therefore, in view of Wood '934, it would have been obvious to one of ordinary skill at the time the invention was made to return the plurality of processed work units from the at least one compute node to the at least one source in the order it was distributed in order to take the order-dependence of the data files into account (as stated in Wood '934 paragraphs [0030]-[0034]).

Re claim 16: The teachings of McIntyre '478 in view of Wood '934 are disclosed above.

However, McIntyre '478 fails to teach the computer readable medium, wherein at least one of the plurality of work units is a control work unit that includes control commands for the at least one compute node.

However, this is well known in the art as evidenced by Wood '934. Wood '934 discloses the computer readable medium, wherein at least one of the plurality of work units is a control work unit that includes control commands for the at least one compute node (i.e. the segment file (22) includes all PDL commands necessary to interpret the associated segment, which includes everything in the global data file associated with the segment. The PDL commands are for commanding the actions of the PDL processors (38), considered as the compute node; see fig. 1; paragraphs [0025]-[0030]).

Therefore, in view of Wood '934, it would have been obvious to one of ordinary skill at the time the invention was made to have at least one of the plurality of work units as a control work unit that includes control commands for the at least one compute node

in order to have a data file include PDL commands necessary to interpret a segment related to an input data stream (as stated in Wood '934 paragraphs [0026]-[0030]).

Re claim 17: The teachings of McIntyre '478 in view of Wood '934 are disclosed above.

However, McIntyre '478 fails to teach the computer readable medium, wherein the processing step (b) includes (b1) executing the control commands in the control work unit.

However, this is well known in the art as evidenced by Wood '934. Wood '934 discloses the computer readable medium, wherein the processing step (b) includes (b1) executing the control commands in the control work unit (i.e. the PDL processors (38) is commanded by the data file (22) with the PDL commands in order to interpret the associated segment related to the data file (22). Therefore, the PDL executes the control commands of the data file (22) in the segment, considered as the work unit; see fig. 1; paragraphs [0025]-[0030]).

Therefore, in view of Wood '934, it would have been obvious to one of ordinary skill at the time the invention was made to have the processing step include executing the control commands in the control work unit in order to interpret the associated segment with the data file that has the control commands (as stated in Wood '934 paragraphs [0026]-[0030]).

Re claim 18: The teachings of McIntyre '478 in view of Wood '934 are disclosed above.

McIntyre '478 discloses the computer readable medium, wherein the at least one source is instantiated as a dynamic library (i.e. when using a transform to convert incoming data into another form, the printer driver (114) requests for a transform to convert from the incoming PDL into a low-level language for the printer to understand. This is performed by linking the incoming data to the specific personality that will perform the transformation of the data stream to the low-level format. This process is similar to a library with a collection of subprograms used to develop other pieces of information used by the system and provide the function of transformation that is linked to a certain input language in the invention of McIntyre '478; see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25).

Re claim 19: McIntyre '478 discloses a method and apparatus for utilizing multiple versions of a page descriptor language comprising:

a central component for receiving the datastream in a first format (i.e. McIntyre '478 discloses a control driver that receives the datastream in a high-level language, or a first format; see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25);

a plurality of sources in the central component, wherein each of the plurality of sources is associated with at least one transform (i.e. the plurality of personalities, considered as sources, are managed by both the control driver (104) and the boot agent (102). Since the control driver manages the personalities, this can be considered as

having the personalities in the control driver (104) to be managed; see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25); and

at least one compute node coupled to the central component (i.e. the printer driver (114), considered as the compute node, is coupled to the control driver (104); see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25),

wherein the central component instantiates at least one source of the plurality of sources (i.e. the control driver (104) uses the printer driver (114) to create a particular realization of a printer description language, or instantiates, through recognizing the personality in the system, which is analogous to the plurality of sources; see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25), and distributes each of the work units to the at least one compute node, wherein the at least one compute node converts each work unit into a second format (i.e. in McIntyre '478, a plurality of print jobs, considered as work units can be processed. The printer driver (114), considered as the compute node, processes the incoming print jobs by recognizing the type of PDL is input into the system. The printer driver selects a PDL type from a PDL registry (112) to correspond with the incoming data and this PDL type chosen is used to convert the data into a different format, analogous to the second format, which is a low-level data stream; see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25).

However, McIntyre '478 fails to teach parses the datastream into a plurality of work units in the first format.

However, this is well known in the art as evidenced by Wood '934. Wood '934 discloses parses the datastream into a plurality of work units in the first format (i.e. the apparatus and method of Wood '934 divides the PDL data stream to provide a plurality of PDL segments and then to create associated first and second data segments for each of the PDL segments. These segments are in a PDL format, analogous to a first format and the function of parsing is performed by the dividing and interpreting the PDL data stream into a plurality of segments, which the plurality of segments are considered as the plurality of work units; see figs. 1-4; paragraphs [0004]-[0014] and [0025]-[0029]).

Therefore, in view of Wood '934, it would have been obvious to one of ordinary skill at the time the invention was made to parse the datastream into a plurality of work units in the first format in order to divide the PDL data stream to provide a plurality of PDL segments (as stated in Wood '934 paragraph [0014]).

Re claim 20: The teachings of McIntyre '478 in view of Wood '934 are disclosed above. The system of claim 19, wherein each of the at least one compute nodes loads the at least one transform as a dynamic library (i.e. the printer driver (114), considered as the compute node, is able to utilize a personality, analogous to a transform, in interpreting incoming data. The incoming data is analyzed by the control driver (104) and the link between the incoming data and the appropriate personality to use for interpretation is made. The link of the incoming data to a collection of software used to change or provide services to other programs is an example of a dynamic library; see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25) and utilizes

the at least one transforms to convert a work unit in the first format to the second format (i.e. the printer driver (114) utilizes one of the personalities, or transforms, to convert a print job, considered as a work unit, from a high-level language to a low-level language that the printer can understand. This is analogous to converting from a first to a second format; see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25).

Re claim 21: The teachings of McIntyre '478 in view of Wood '934 are disclosed above.

However, McIntyre '478 fails to teach the system of claim 19, wherein the central component further includes: a load balancing mechanism coupled to the at least one source for distributing the plurality of work units to the at least one compute node, wherein the load balancing mechanism generates at least one queue for the plurality of work units and dispatches each work unit from the at least one queue to the at least one compute node in an order received from the at least one source.

However, this is well known in the art as evidenced by Wood '934. Wood '934 discloses

a load balancing mechanism coupled to the at least one source for distributing the plurality of work units to the at least one compute node (i.e. in Wood '934, the segment data files (22) are in a buffer memory (40) with a scheduler process (24), analogous to a load balancing mechanism, that accesses the data files and distributes the data files, analogous to work units, to the PDL processors, which are analogous to the compute nodes; see figs. 1 and 4; paragraphs [0030]-[0034]),

wherein the load balancing mechanism generates at least one queue for the plurality of work units and dispatches each work unit from the at least one queue to the at least one compute node in an order received from the at least one source (i.e. the scheduler process (24) has several data files stored, or queued, in the memory buffers and the data segments are given to the PDL processors from the buffer memories with the order-dependence taken into account. Depending on the order of the data files in the data stream is the order that the data files are given to, or dispatched, to the PDL processors; see figs. 1 and 4; paragraphs [0030]-[0034]).

Therefore, in view of Wood '934, it would have been obvious to one of ordinary skill at the time the invention was made to have a load balancing mechanism coupled to the source for distributing a plurality of work units to the compute node, generating a queue for the work units and dispatching the work units from the queue to the compute node in an order received from the source in order to store data segments is a memory buffer and have a scheduling process able to access the memory buffer and assign data files to PDL processors in an order dependent manner (as stated in Wood '934 paragraphs [0030]-[0034]).

Re claim 22: The teachings of McIntyre '478 in view of Wood '934 are disclosed above. McIntyre '478 discloses the system, wherein the work units processed by the at least one compute node are returned to the at least one source (i.e. in McIntyre '478, the printer driver (114) invokes the personalities to perform the transformation of receiving information. If the system invokes a personality to perform the conversion of input

received data, then the personality aspect of the invention performs the transformation of the PDL and returns this conversion to the printer driver (114). In this case, the personality utilized to perform transformation is considered to be the compute node, which performs the transformation, and the printer driver (114) is considered to be the source since the data was received by the printer driver (114) and sent to the personality (106) from the printer driver (114) and sent back to the printer driver (114) in order to be output by the output device; see figs. 1 and 5; col. 2, lines 18-30, col. 3, lines 1-67, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25).

However, this is well known in the art as evidenced by Wood '934. Wood '934 discloses in the order in which the work units were dispatched (i.e. Wood '934 reveals that the prior art involves a process of combining a data stream portion in the order of the original PDL data stream. In Wood '934, when the PDL processes the data, the scheduling process (24) accounts for the order-dependence of the data files in the buffer memories; see fig. 4; paragraphs [0030]-[0037]).

Therefore, in view of Wood '934, it would have been obvious to one of ordinary skill at the time the invention was made to the work units processed by the at least one compute node are returned to the at least one source in the order in which the work units were dispatched in order to take the order-dependence of the data files into account (as stated in Wood '934 paragraphs [0030]-[0034]).

Re claim 23: The teachings of McIntyre '478 in view of Wood '934 are disclosed above.

However, McIntyre '478 fails to teach the system, wherein at least one of the plurality of work units is a control work unit that includes commands for the at least one compute node.

However, this is well known in the art as evidenced by Wood '934. Wood '934 discloses the system, wherein at least one of the plurality of work units is a control work unit that includes commands for the at least one compute node (i.e. the segment file (22) includes all PDL commands necessary to interpret the associated segment, which includes everything in the global data file associated with the segment. The PDL commands are for controlling the actions of the PDL processors (38), considered as the compute node; see fig. 1; paragraphs [0025]-[0030])

Therefore, in view of Wood '934, it would have been obvious to one of ordinary skill at the time the invention was made to have a system, wherein at least one of the plurality of work units is a control work unit that includes commands for the at least one compute node in order to have a data file include PDL commands necessary to interpret a segment related to an input datastream (as stated in Wood '934 paragraphs [0026]-[0030]).

Re claim 24: The teachings of McIntyre '478 in view of Wood '934 are disclosed above.

However, McIntyre '478 fails to teach the system, wherein the at least one compute node processes the control work unit by executing the command.

However, this is well known in the art as evidenced by Wood '934. Wood '934 discloses the system, wherein the at least one compute node processes the control

work unit by executing the command (i.e. the PDL processors (38) is commanded by the data file (22) with the PDL commands in order to interpret the associated segment related to the data file (22). Therefore, the PDL executes the control commands of the data file (22) in the segment, considered as the work unit; see fig. 1; paragraphs [0025]-[0030]).

Therefore, in view of Wood '934, it would have been obvious to one of ordinary skill at the time the invention was made to have the system, wherein at least one compute node processes the control work unit by executing the command in order to interpret the associated segment with the data file that has the control commands (as stated in Wood '934 paragraphs [0026]-[0030]).

Re claim 25: The teachings of McIntyre '478 in view of Wood '934 are disclosed above. McIntyre '478 discloses the system, wherein the at least one source is instantiated as a dynamic library (i.e. when using a transform to convert incoming data into another form, the printer driver (114) requests for a transform to convert from the incoming PDL into a low-level language for the printer to understand. This is performed by linking the incoming data to the specific personality that will perform the transformation of the data stream to the low-level format. This process is similar to a library with a collection of subprograms used to develop other pieces of information used by the system and provide the function of transformation that is linked to a certain input language in the invention of McIntyre '478; see figs. 1 and 3; col. 2, lines 18-30, col. 4, lines 11-66, col. 5, lines 160 and col. 6, lines 1-25).

Conclusion

3. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

4. Vennekens '711 (US Pat No 5652711) discloses a system that parses a data stream into a plurality of PDL data stream segments in a high-level language and each of these segments are stored in a FIFO queue before the segments are output to a sub-process where the segments are converted to a low-level language. Load balancing is performed in regards to the segments in the system.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Chad Dickerson whose telephone number is (571)-270-1351. The examiner can normally be reached on Mon. thru Thur. 9:00-6:30 Fri. 9:00-5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Aung Moe can be reached on (571)- 272-7314. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

CD/
Chad Dickerson
July 17, 2007

AUNG S. MOE
SUPERVISORY PATENT EXAMINER
7/19/07